

Exploring Python - Lab 1 - optional extension



1. Turtles! (allow 30-45 minutes)

“Turtle” is a python feature like a drawing board, which lets you command a turtle to draw all over it. You can use methods like `turtle.forward(...)` and `turtle.left(...)` which can move the turtle around.

We will use turtle quite regularly in the first few weeks of the module to illustrate some basic concepts. It isn't related explicitly to cybersecurity, but a bit of fun, and a great tool to practice problem solving and logical thinking too.

Before you can use turtle, you have to **import** the corresponding python module (package, library).

```
>>> import turtle
```

(if you get no error message, the import has worked and you can continue)

```
>>> turtle.forward(100)
```

You should see a new window open with a drawing canvas. Keep this visible and watch what happens as you execute the following commands.

```
>>> turtle.left(90)
```

```
>>> turtle.forward(50)
```

```
>>> turtle.shape("turtle")
```

```
>>> turtle.left(45)
```

```
>>> turtle.forward(100)
```

Q: what does the `turtle.forward()` method do? What does the method take as parameter?

Q: what does the `turtle.left()` method do? What does the method take as parameter?

Q: what do you expect the corresponding methods `turtle.backward()` and `turtle.right()` to do?

Note

Want to start fresh? You can type **`turtle.reset()`** to clear the drawing that your turtle has made so far.

```
>>> turtle.reset()
```

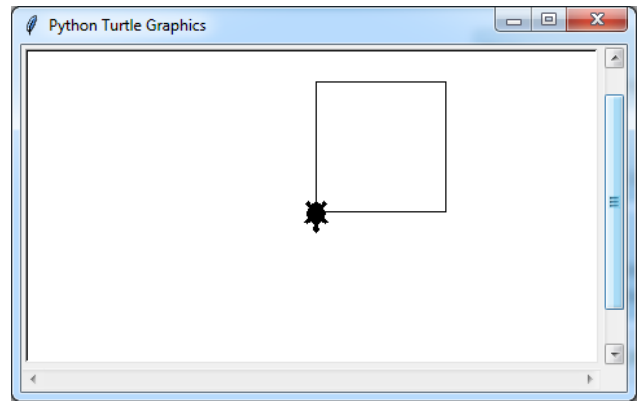
You should have a clear canvas again now, with just the turtle showing.

Now draw a square where each side is 100 long.
Your drawing should look like this:

Hint: you need only the methods `.forward()` and `.left()`.
Remember you can use Alt-P to repeat previous commands.

(If you make a mistake use

`>>> turtle.undo()` #to retrace your steps.)



You will have seen that the simplest way is to repeat

```
>>> turtle.forward(100)
```

```
>>> turtle.left(90)
```

four times.

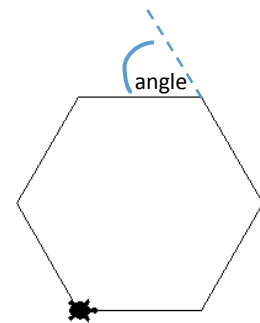
Before moving on, read

https://opentechschoo.github.io/python-beginners/en/simple_drawing.html and do the small exercises there.

Q: How many sides and corners does a hexagon have?

Q: To draw a hexagon instead of a square, what angle
do you need at each corner?

(hint: a full circle is 360 degrees, divide by the number of corners)



All in, that would need 12 statements to draw a hexagon,
6 for the sides and 6 for the corners turned. This is where loops come in - they are ideal where you
need to repeat the same statement(s) many times.

A "for" loop to draw the square would be:

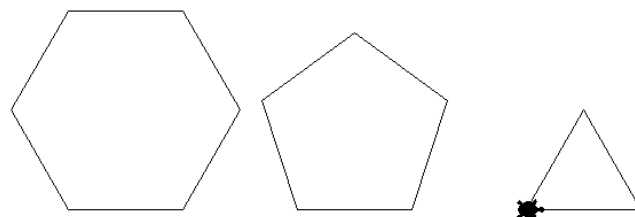
```
>>> for i in range(4):  
    turtle.forward(100)  
    turtle.left(90)
```

Here `range(4)` is used to cycle through the loop 4 times.

Reset the drawing canvas then try out the above loop.

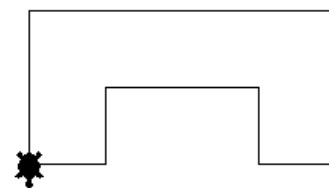
To draw a hexagon, pentagon, or triangle, you just need to replace the 4 and the 90 in the loop by appropriate values. Try it out!

shape	number of sides	angle at each corner
square	4	90
hexagon		
pentagon		
triangle		



2. More Turtle exercises

- (a) Draw this diagram, let's call it a gate. The overall width should be 200 and the overall height 100. You need to work out the lengths of all the lines accordingly. In addition to `.forward()` and `.left()` you will also need the `.right()` method.



- (b) Redraw the gate but this time fill it in blue. You will need three additional methods: `.color('blue')`, `.begin_fill()` and `.end_fill()`.



- (c) Draw this diagram. Use 100 for the sides of the inner square. The sides of the outer square are then $100 \times 2 \times 0.5$ long (worked out using Pythagoras' theorem). Divide this number by 2 to draw only half of a side of the outer square. It is fine to give a calculation as the argument to the `.forward()` method, e.g. `turtle.forward(100*2**0.5/2)`. Remembering that a straight line is 180 degrees, to get the acute angles in the diagram you will need to turn $180-45=135$ degrees.

